

**AMENDMENTS TO THE CLAIMS**

*Please amend the Claims as follows:*

**1. (Currently Amended)** A method for validating a fragment of a structured document comprising steps of:

- a. compiling an XML schema definition,
- b. storing said XML schema definition,
- c. receiving as input said stored XML schema definition and a fragment of a structured document into a runtime validation engine, and
- d. outputting a validation pass or fail on the basis of said input.

**2. (Original)** A method of validating a fragment of a structured document, as per claim 1, wherein said structured document is an XML document.

**3. (Original)** A method for validating a fragment of a structured document, as per claim 1, wherein said runtime validation engine is comprised of a generic parser and a runtime schema validation parser.

**4. (Original)** A method for validating a fragment of a structured document, as per claim 3, wherein said generic parser is a generic XML parser.

**5. (Currently Amended)** A method of preparing a fragment of a structured document for validation comprising steps of:

- a. locating a start state for said validation process from a type-mapping table,
- b. obtaining a token from said structured document fragment,
- c. determining whether said token is of element type said structured document fragment is to be validated against,
- d. checking whether said token signifies end of said structured document, and
- e. returning a validation success or a validation failure, based on said determining and checking steps.

6. **(Original)** A method of preparing a fragment of a structured document, as per claim 5, wherein said structured document is an XML document.

7. **(Original)** A method of preparing a fragment of a structured document, as per claim 5, wherein said token is either an element type name or an attribute name.

8. **(Original)** A method of preparing a fragment of a structured document, as per claim 5, wherein if in said determining step it is determined that said token is not of said element type, returning a validation failure, else repeating process from said obtaining step.

9. **(Original)** A method of preparing a fragment of a structured document, as per claim 5, wherein if in said checking step it is determined that said token signifies end of said structured document, said validation process terminates.

**10. (Original)** A method of preparing a fragment of a structured document, as per claim 5, wherein said validation process is repeated from said obtaining step until said validation process returns a validation failure or it is determined in said checking step that said obtained token signifies end of said structured document and said validation process terminates.

**11. (Currently Amended)** A method of constructing a type-mapping table comprising steps of:

- a. building a type hierarchy ordered tree from a structured document schema,
- b. supplying input to an element validation module,
- c. creating a type-mapping table entry for a current element type in said structured document schema,
- d. traversing said type hierarchy ordered tree, and
- e. populating a type-mapping table with type-mapping entries created in said creating step.

**12. (Original)** A method of constructing a type-mapping table, as per claim 11, wherein said structured document schema is an XML schema.

**13. (Original)** A method of constructing a type-mapping table, as per claim 11, wherein said method takes as input an AAE.

**14. (Original)** A method of constructing a type-mapping table, as per claim 13, wherein said AAE is comprised of an annotation hierarchy and an automaton encoding.

**15. (Original)** A method of constructing a type-mapping table, as per claim 11, wherein said data structures and variables are comprised of a token array, a variable holding the index of the last token received, and a variable holding the index of start token received.

**16. (Original)** A method of constructing a type-mapping table, as per claim 11, wherein said type-mapping table entry for said element type is formed by supplying a start token from an annotation record to an element validation module.

**17. (Original)** A method of constructing a type-mapping table, as per claim 11, wherein said element validation module is reset after each entry is created for each element type.

**18. (Original)** A method of constructing a type-mapping table, as per claim 11, wherein said data structures are initialized and said variables are set to zero after an entry has been created for each element type.

**19. (Original)** A method of constructing a type-mapping table, as per claim 11, wherein said type-mapping table entries are comprised of a result path to current element type, a current element type, an annotation record for current element type, and a current state.

**20. (Original)** A method of constructing a type-mapping table, as per claim 11, wherein said process is repeated for each global element child type.

**21. (Original)** A method of constructing a type-mapping table, as per claim 11, wherein said traversing step further comprises steps of:

- a. determining whether an entry has been created for all element types in said schema,
- b. appending a start token of a current sub-element type to a token array data structure,
- c. incrementing an environment variable representing an index for a last token
- d. supplying said token to said element validation module,
- e. creating an entry for said current sub-element type in said type-mapping table, and
- f. updating data structures and variables.

**22. (Original)** A method of constructing a type-mapping table, as per claim 21, wherein said traversing step takes as input an AAE, said current element type, and said data structures and variables.

**23. (Original)** A method of constructing a type-mapping table, as per claim 21, wherein said entry is comprised of a result path for said current sub-element type, an element type name for said current sub-element type, an annotation record for said current sub-element type, and a current state.

**24. (Original)** A method of constructing a type-mapping table, as per claim 21, wherein if said current sub-element type is a reference to a global element type, said result path is a union of the path from root of said schema to said current sub-element type and the result path in a type-

mapping entry in said type-mapping table of said referenced global type; otherwise said result path is the path from root of said schema to said sub-current element type.

**25. (Original)** A method of constructing a type-mapping table, as per claim 24, wherein said updating step further comprises steps of:

- a. setting a current index variable equal to an index variable representing an index of last token and
- b. pushing an annotation record for said current sub-element type and said current index of said token array onto a temp stack.

**26. (Original)** A method of constructing a type-mapping table, as per claim 21, wherein said traversing step is recursively performed until type-mapping entries are created for all sub-element types descending from said current element type.

**27. (Original)** A method of constructing a type-mapping table, as per claim 21, wherein if it is determined in said determining step that an entry has been created for all element types in said schema; an end token is appended to said token array, said token is supplied to said element validation module, an annotation record for said current sub-element type is obtained from said temp stack, all tokens from a subset of indices within said token array are supplied to said element validation module, and said process terminates.

**28. (Currently Amended)** An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements the validation a fragment of a structured document, said medium comprising:

- a. computer readable program code compiling an XML schema definition,
- b. computer readable program code storing said XML schema definition,
- c. computer readable program code receiving as input said stored XML schema definition and a fragment of a structured document into a runtime validation engine, and
- d. computer readable program code outputting a validation pass or fail on the basis of said input.

**29. (Currently Amended)** An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements the construction a type-mapping table, said medium comprising:

- a. computer readable program code building a type hierarchy ordered tree from a structured document schema,
- b. computer readable program code supplying input to an element validation module,
- c. computer readable program code creating a type-mapping table entry for a current element type in said structured document schema,
- d. computer readable program code traversing said type hierarchy ordered tree, and
- e. computer readable program code populating a type-mapping table with type-mapping entries created in said creating module.